# Pomoc AppleScript

Podstawy

## Omówienie AppleScript

W miarę korzystania z Maca zauważysz, że regularnie powtarzasz pewne czynności. Ich wykonywanie można zautomatyzować, pisząc odpowiedni skrypt lub program AppleScript.

AppleScript jest potężnym oraz uniwersalnym językiem skryptowym wbudowanym w OS X. Można go używać do tworzenia skrótów, automatyzowania powtarzalnych zadań lub nawet do tworzenia własnych programów, które pozwolą zaoszczędzić czas. AppleScript pozwala kontrolować programy skryptowalne.

Skrypt AppleScript składa się z poleceń pisanych w języku zbliżonym do angielskiego. Polecenia te wykonują określone czynności z różnymi obiektami. Kilka przykładowych czynności to „open” (otwórz), „close” (zamknij) lub „get the name of” (pobierz nazwę). Obiektami mogą być programy, okna lub ustawienia w preferencjach.

Skrypt może kontrolować jeden lub kilka skryptowalnych programów. Aby program był skryptowalny, musi zawierać definicje wyrażeń, których będzie można użyć w skryptach. Wiele programów dołączonych do systemu OS X jest skryptowalnych.

Skrypty mogą także kontrolować programy w sieci, można więc automatyzować zadania na wielu komputerach.

Z AppleScript można się spotkać nawet o tym nie wiedząc. Wiele funkcji w systemie oraz poszczególnych programach obsługiwanych jest przez skrypty AppleScript. Na przykład, w oknie Przeglądarki pomocy można klikać w łącza. Kliknięcie w łącze otwierające program lub plik powoduje uruchomienie skryptu, który wykonuje tę czynność.

**PODOBNE TEMATY**

Jak działa AppleScript
Edytor AppleScript
Parę słów o języku AppleScript
Zasoby AppleScript w Internecie
Środowisko programistyczne AppleScript Studio

Podstawy

## Edytor AppleScript

Edytor AppleScript to łatwy w obsłudze program do otwierania, pisania i nagrywania skryptów AppleScript.

Oto kilka funkcji programu Edytor AppleScript:

- posiada ujednolicony, łatwy w użyciu interfejs,
- pozwala na wyszukiwanie i zastępowanie tekstu w skryptach,
- automatycznie zawija wiersze z długimi instrukcjami,
- pozwala cofać wprowadzone zmiany,
- udostępnia menu skrótów, przyspieszające pisanie skryptów,

- obsługuje usługi programów,
- okno biblioteki pozwala na dostęp do skryptów programów,
- Asystent skryptu pomaga w pisaniu instrukcji,
- historia wyników i historia dziennika zdarzeń pozwalają na analizowanie działania skryptu.

**PODOBNE TEMATY**

Parę słów o języku AppleScript
Środowisko programistyczne AppleScript Studio

---

**Podstawy**

## Tworzenie skryptów AppleScript

Nie trzeba znać się na programowaniu, aby korzystać z AppleScript. Można korzystać z niego nie wpisując nawet jednego polecenia skryptu. Do systemu OS X dołączone są setki przydatnych skryptów AppleScript, a kolejne tysiące innych dostępne są w sieci.

W istniejących skryptach można dokonywać zmian, aby dostosować je do własnych potrzeb.

Najlepiej jednak jest tworzyć własne skrypty od podstaw. Umiejętność pisania skryptów pozwoli tworzyć cenne narzędzia, dzięki którym będzie można zaoszczędzić czas i zwiększyć wydajność pracy.

Możesz modyfikować istniejące skrypty lub tworzyć nowe, używając do tego celu Edytora AppleScript. Program ten zapewnia proste środowisko umożliwiające pisanie skryptów. Wszystkie przykłady dostępne w pomocy AppleScript zostały przygotowane właśnie za pomocą Edytora AppleScript.

---

- Otwórz Edytor AppleScript

  W oknie Edytora AppleScript wpisz poniższy tekst.

  ```
  say "AppleScript wita!"
  ```

  Kliknij w Uruchom.

---

**PODOBNE TEMATY**

Edytor AppleScript
Środowisko programistyczne AppleScript Studio

---

**Podstawy**

## Jak działa AppleScript

AppleScript działa przez wysyłanie komunikatów, tzw. zdarzeń Apple (Apple Events). W trakcie wykonywania skryptu AppleScript wysyła te zdarzenia do systemu i programów, które mają wykonać określone polecenia.

W większości przypadków programy lub system będą zwracać wynik działania do AppleScript. Na przykład, jeśli skrypt wyśle polecenie do programu, aby uzyskać wartość jakiegoś ustawienia, program w odpowiedzi zwróci tę wartość. Jeśli polecenie nie może zostać wykonane, program lub AppleScript może zwrócić komunikat o błędzie. Zwrócone wyniki można obejrzeć w Edytorze AppleScript.

Znajomość sposobu działania zdarzeń Apple nie jest niezbędna. AppleScript wykonuje to wszystko automatycznie w tle. Ty musisz tylko napisać i uruchomić skrypt.

**PODOBNE TEMATY**

Omówienie AppleScript
Parę słów o języku AppleScript

---

**Podstawy**

## Parę słów o apletach

Skrypt zachowany jako program nazwany jest inaczej programem AppleScript lub apletem. Aplety uruchamiają się automatycznie po otwarciu i wykonują polecenia zawarte w skrypcie. Mogą to być zarówno proste skrypty, wykonujące tylko kilka poleceń, jak i większe programy, składające się z tysięcy wierszy.

Każdy aplet można zachować w postaci edytowalnej lub jako plik tylko do uruchomienia. Edytowalny aplet zawiera tekst skryptu, dzięki czemu w dowolnej chwili można otworzyć go i dokonać w nim zmian. Wystarczy w tym celu przeciągnąć jego ikonę do ikony Edytora AppleScript. Po zachowaniu skryptu jako plik tylko do uruchomienia, tekst skryptu jest z niego usuwany. Sprawia to, że w skrypcie takim nie można dokonać żadnych zmian.

Jeśli aplet został zachowany jako plik edytowalny, można otworzyć go w Edytorze AppleScript lub dowolnym innym edytorze AppleScript. Każdy aplet musi zawierać instrukcję on run, uruchamianą po otwarciu apletu.

- Poniższy przykład ilustruje użycie instrukcji on run.

```
on run
        --first command
        --second command
end run
```

**PODOBNE TEMATY**

Parę słów o dropletach
Typy plików AppleScript

---

**Podstawy**

## Parę słów o skompilowanych skryptach

Skompilowany skrypt to plik AppleScript w takiej formie, która pozwala na jego edycję w Edytorze AppleScript lub w innym edytorze AppleScript.

Po otwarciu skompilowanego skryptu jego tekst pojawi się w oknie Edytora AppleScript, ale polecenia skryptu nie zostaną wykonane. Pozwoli to na edycję skryptu bez jego uruchamiania. Po dokonaniu edycji można skompilować skrypt, co zamieni jego tekst na kod AppleScript. Skompilowany skrypt można uruchomić. Rozszerzenie nazwy skompilowanego skryptu to .scpt.

Po zachowaniu skompilowanego skryptu na dysku, zapamiętywany jest także jego tekst, razem z kodem. Możesz w ten sposób ponownie edytować skompilowane skrypty.

**PODOBNE TEMATY**

Parę słów o apletach
Parę słów o dropletach
Typy plików AppleScript

---

Podstawy

## Parę słów o dropletach

Droplet AppleScript to skrypt AppleScript, uruchamiany po przeciągnięciu do jego ikony plików lub katalogów. Skrypt ten wykonuje czynności z przeciągniętymi rzeczami.

Podobnie jak aplety, droplety mogą być zachowywane jako pliki edytowalne lub pliki tylko do uruchomienia. Edytowalny droplet zawiera tekst skryptu, dzięki czemu w dowolnej chwili można otworzyć go i dokonać w nim zmian. Wystarczy w tym celu przeciągnąć ikonę dropletu do ikony Edytora skryptów. Po zachowaniu skryptu jako plik tylko do uruchomienia, tekst skryptu jest z niego usuwany. Sprawia to, że w skrypcie takim nie można dokonać żadnych zmian.

Każdy droplet musi zawierać instrukcję „on open”, uruchamianą po jego otwarciu. Instrukcja „on open” posiada parametr zawierający listę rzeczy przeciągniętych do dropletu. Część skryptu dropletu sprawdza listę, a inna część działa na obiektach. Jeśli droplet ma obsługiwać więcej niż pojedyncze rzeczy, musi zawierać pętlę iterującą przez wszystkie elementy listy, która będzie wykonywać określone czynności z każdym z nich.

- Poniższy przykład ilustruje użycie instrukcji „on open” w droplecie razem z listą parametrów.

```
on open (theItemsDropped)
        --first command
        --second command
end open
```

**PODOBNE TEMATY**

Parę słów o apletach
Parę słów o skompilowanych skryptach
Typy plików AppleScript

---

Podstawy

## Korzystanie ze słowników AppleScript

Ze słownika AppleScript należącego do danego programu możesz się dowiedzieć, jakie polecenia i obiekty rozpoznaje dany program. Słownik taki zawiera zbiór terminów AppleScript zrozumiałych dla danego programu.

Słownik AppleScript dowolnego skryptowalnego programu możesz przeglądać za pomocą Edytora AppleScript zaznaczając program w oknie biblioteki lub przeciągając jego ikonę do ikony Edytora AppleScript.

Terminy w słowniku są zorganizowane w grupy zwane zestawami (ang. suites). Polecenia i obiekty są napisane pogrubioną czcionką.

1. W Edytorze AppleScript wybierz polecenie menu Okno > Biblioteka.
2. Kliknij dwukrotnie w nazwę programu w oknie Biblioteka.

3. Zaznacz zestaw.

   Zostaną wyświetlone wyrażenia należące do zestawu wraz z opisami.

---

Język AppleScript jest podzielony na różne kategorie wyrażeń:

- Zestawy: Grupy powiązanych wyrażeń i terminów W słowniku Edytora AppleScript zestawy widnieją w pierwszej kolumnie, oznaczone pomarańczową literą S.
- Polecenia: powodujące wykonanie czynności lub zwrócenie wyniku. Polecenia, zwane też zdarzeniami, widnieją w drugiej kolumnie i są oznaczone niebieską literą C (od ang. command).
- Klasy: Skryptowalne obiekty. Opis klasy obejmuje właściwości i elementy należące do danej klasy. Klasy pojawiają się w drugiej kolumnie i są oznaczone literą C w kolorze lawendy (od ang. classes).
- Właściwości: Obiekty zawierające określone wartości. Właściwości pojawiają się w trzeciej kolumnie, oznaczonej literą P w kolorze lawendy (od angielskiego Properties).
- Elementy: Obiekty zawarte w innych obiektach. Na przykład, akapit może zawierać wiele elementów, czyli słów. Elementy mogą pojawić się w trzeciej kolumnie i są oznaczone pomarańczową literą E.

**PODOBNE TEMATY**

Definicje klas
Definicje poleceń
Sterowanie programami przy użyciu AppleScript

**OTWÓRZ**

Przewodnik języka AppleScript Przewodnik po poleceniach
Przewodnik języka AppleScript Przewodnik po klasach

---

**Podstawy**

## Rozwiązywanie problemów

Jeśli masz problem z korzystaniem z AppleScript, przejrzyj poniższe tematy pomocy, aby znaleźć informacje pozwalające na rozwiązanie problemu.

Gdy po kliknięciu w Anuluj w oknie wyboru z listy pojawi się błąd

Gdy skrypt czynności katalogów został nieoczekiwanie uruchomiony

Gdy skrypt nie rozpoznaje dysku

---

**Polecenia podstawowe**

## Definicje klas

Definicja klasy opisuje właściwości i elementy skryptowalnego obiektu. Klasa może być przyrównana do rzeczownika w zdaniu, podczas gdy definicja klasy przypomina definicję rzeczownika w słowniku.

Typowa definicja klasy w słowniku AppleScript zaczyna się od pogrubionego wyrażenia, po którym następuje jego krótki opis (w tym samym wierszu).

Dalej wyświetlana jest lista wszystkich elementów lub obiektów, należących do skryptowanego obiektu. Nie jest ona obecna przy wszystkich definicjach klas, ponieważ nie wszystkie skryptowalne obiekty zawierają inne skryptowalne obiekty.

Definicja klasy kończy się listą właściwości skryptowalnego obiektu i ich wartościami.

Opis właściwości rozpoczyna się nazwą tej właściwości, po której następuje:

- opis formatu danych, w którym przechowywana jest wartość, np. integer (liczba całkowita), point (lista dwóch liczb), string (łańcuch tekstowy) oraz Boolean (prawda lub fałsz, czyli true lub false),
- opcjonalny wskaźnik typu, informujący o tym, czy wartość właściwości jest tylko do odczytu (nie można jej zmieniać),
- krótki opis właściwości.

Jeśli obiekt występuje w liczbie mnogiej, można odwoływać się w skrypcie do kilku takich obiektów.

Jeśli obiekt posiada listę elementów, zawiera w sobie inne obiekty. Nazwa elementu, której można użyć w skrypcie, wyświetlana jest pogrubioną czcionką. Sposób odnoszenia się do obiektu opisany jest zwykłą czcionką.

Obiekty mogą mieć właściwości. Nazwy właściwości wyświetlane są pogrubioną czcionką. Typy właściwości (np. string, czyli łańcuch tekstowy) opisane są zwykłą czcionką. Właściwości mogą być stałymi lub wartościami z listy stałych. Skrypt może pobrać wartość właściwości oraz zmienić wartość właściwości, chyba że dana właściwość jest typu read only (tylko do odczytu).

**PODOBNE TEMATY**
Definicje poleceń
Definicje obiektów
Polecenia AppleScript
Korzystanie ze słowników AppleScript

**OTWÓRZ**
Przewodnik języka AppleScript Przewodnik po klasach

---



## Definicje poleceń

Polecenia to terminy należące do języka AppleScript, powodujące wykonanie określonych czynności. W słowniku AppleScript polecenia wyświetlane są w środkowej kolumnie i są oznaczone niebieską literą C (od ang. commands). Definicja polecenia zawiera skrótowy opis jego działania oraz jego składnię. Składnia to sposób zapisu polecenia w skrypcie. Może się ona składać z kilku części.

Pogrubione słowa to te, które używane są w instrukcji. Trzeba wpisać je dokładnie w takiej samej formie, w jakiej występują w słowniku.

Słowa w kolorze niebieskim to informacje podawane przez Ciebie. Może to być np. nazwa obiektu lub zmiennej, zdefiniowanej przez użytkownika.

Niektóre polecenia zawierają opcjonalne fragmenty, ujęte w nawias. Nie trzeba ich wpisywać w instrukcji.

**PODOBNE TEMATY**
Definicje klas
Korzystanie ze słowników AppleScript
Definicje zdarzeń

Definicje obiektów

**OTWÓRZ**

Przewodnik języka AppleScript Przewodnik po poleceniach

---

Polecenia podstawowe

## Definicje zdarzeń

Definicja zdarzenia opisuje polecenie (czasownik), używane do skierowania czynności na skryptowany obiekt. Zdarzenia to czynności (czasowniki), które obiekty mogą wykonywać lub w nich uczestniczyć.

Definicja zdarzenia rozpoczyna się zwykle od pogrubionego wyrażenia, za którym następuje jego krótki opis (w tym samym wierszu).

Dalej podana jest lista wszelkich parametrów, używanych przy wykonywaniu polecenia. Parametry znajdujące się w nawiasach są opcjonalne, natomiast parametry nie ujęte w nawias są wymagane.

Opis każdego parametru zaczyna się od jego nazwy, za którą następuje opis formatu danych używanego przez parametr oraz krótki opis samego parametru.

Definicja polecenia zawiera krótki opis działania polecenia (podkreślony) oraz jego składni. Składnia to sposób zapisu polecenia w skrypcie. Może się ona składać z kilku części.

Pogrubione słowa to te, które używane są w instrukcji. Trzeba wpisać je dokładnie w takiej samej formie, w jakiej występują w słowniku.

Słowa zapisane zwykłą czcionką oznaczają informacje podawane przez użytkownika. Może to być np. nazwa obiektu lub zmiennej, zdefiniowanej przez użytkownika.

Niektóre polecenia zawierają opcjonalne fragmenty, ujęte w nawias. Nie trzeba ich wpisywać w instrukcji.

Polecenia dotyczą obiektów. Definicja obiektu zawiera opis składni służącej do identyfikacji obiektu, włączając w to jego nazwę i krótki opis podkreśloną czcionką. Klasa obiektu określa jego charakterystykę, np. polecenia, których można użyć w stosunku do danego obiektu.

**PODOBNE TEMATY**

Zdarzenia bazy danych
Definicje klas
Definicje poleceń
Definicje obiektów
Jak działa AppleScript

---

Polecenia podstawowe

## Definicje obiektów

Polecenia dotyczą obiektów. Definicja obiektu w słowniku AppleScript zawiera opis składni służącej do identyfikacji obiektu, w tym jego nazwę i krótki opis (podkreślony tekst). Klasa obiektu określa jego charakterystykę, np. polecenia, których można użyć w stosunku do danego obiektu.

Jeśli obiekt występuje w liczbie mnogiej, można odwoływać się w skrypcie do kilku takich obiektów.

Jeśli obiekt posiada listę elementów, zawiera w sobie inne obiekty. Nazwa elementu, której można użyć w skrypcie, wyświetlana jest pogrubioną czcionką. Sposób odnoszenia się do obiektu opisany jest zwykłą czcionką.

Obiekty mogą mieć właściwości. Nazwy właściwości zapisane są pogrubioną, niebieską czcionką. Typy właściwości (np. string czyli łańcuch tekstowy) opisane są zwykłą czcionką w kolorze niebieskim. Właściwości mogą być stałymi lub wartościami z listy stałych. Skrypt może pobrać wartość właściwości oraz zmienić wartość właściwości, chyba że dana właściwość jest oznaczona jako read only (tylko do odczytu).

**PODOBNE TEMATY**

Sterowanie programami przy użyciu AppleScript
Definicje poleceń
Definicje klas
Sposób przekazywania poleceń obiektom
Korzystanie ze słowników AppleScript

**OTWÓRZ**

Przewodnik języka AppleScript Przewodnik po klasach
Przewodnik języka AppleScript Właściwości

---



## Instrukcje if

Instrukcja warunkowa if to polecenie, które jest wykonywane tylko wtedy, gdy są spełnione określone warunki. Każda instrukcja warunkowa zawiera wyrażenie logiczne, którego wynikiem może być prawda lub fałsz. Polecenia zawarte w instrukcji warunkowej wykonywane są tylko wtedy, gdy wyrażenie to jest prawdziwe. Instrukcja warunkowa kończy się poleceniem end if. Na przykład:

```
if amountReceived > 100 then
        set memberStatus to "special"
end if
```

W powyższym przykładzie, gdy wyrażenie logiczne jest prawdziwe (czyli amountReceived jest większe od 100), wykonywane są polecenia wewnątrz bloku if. Jeśli wyrażenie to nie jest prawdziwe, polecenia te nie są wykonywane.

W instrukcji warunkowej można użyć także polecenia else (w przeciwnym razie). Jest to tzw. instrukcja if-then-else. Na przykład:

```
if amountReceived > 100 then
        set memberStatus to "special"
else
        set memberStatus to "general"
end if
```

W powyższym przykładzie, gdy pierwsze wyrażenie jest prawdziwe (czyli amountReceived jest większe od 100), wykonywane są polecenia w pierwszym bloku. Jeśli pierwsze wyrażenie logiczne jest fałszywe, wykonywane jest polecenie znajdujące się w drugim bloku (po instrukcji else).

Instrukcje warunkowe można zagnieżdżać, aby obsłużyć różne warunki.

Więcej informacji o instrukcjach warunkowych można znaleźć w Przewodniku języka AppleScript.

**PODOBNE TEMATY**

Instrukcje repeat
Definicje poleceń
Ustawianie zmiennych w AppleScript
Wartości stałe

**OTWÓRZ**

Przewodnik języka AppleScript - If

---

Polecenia podstawowe

# Instrukcje repeat

Instrukcji repeat można używać do wielokrotnego wykonywania zadania w zależności od spełnienia pewnych warunków. Instrukcja repeat powtarza grupę poleceń do momentu spełnienia określonych warunków. Instrukcje repeat można zagnieżdżać, aby obsłużyć różne warunki.

W AppleScript dostępnych jest szereg instrukcji repeat, sprawdzających różne warunki.

### repeat (forever)

Powtarza instrukcje w pętli. Do opuszczenia pętli służy instrukcja exit.

```
repeat
        -- statement
        -- statement
        if (x = 5) then exit repeat
end repeat
```

### repeat (liczba) times

Instrukcje w pętli powtarzane są określoną liczbę razy.

```
repeat 5 times
        -- statement
        -- statement
end repeat
```

### repeat while

Powtarza instrukcje w pętli, gdy określony warunek jest prawdziwy.

```
set x to 1
repeat while (x < 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

### repeat until

Powtarza instrukcje w pętli, dopóki określony warunek jest prawdziwy.

```
set x to 1
repeat until (x = 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

---

**repeat with (zmienna) from (startValue) to (stopValue)**

Ustawia wartość zmiennej na startValue, a następnie zwiększa ją o jeden przy każdym powtórzeniu pętli. Powtarzanie kończy się po osiągnięciu przez zmienną wartości stopValue.

```
repeat with x from (5) to (25)
        -- statement
        -- statement
end repeat
```

---

**repeat with (zmienna) in (lista)**

Ustawia zmienną na pierwszy element z listy. Przy każdym powtórzeniu pętli zmienna przyjmuje wartość kolejnego elementu z listy. Pętla kończy się po użyciu wszystkich elementów.

```
set myList to {234, 342, 681, 874}
repeat with x in myList
        -- statement
        -- statement
end repeat
```

---

Więcej informacji o pętlach można znaleźć w Przewodniku języka AppleScript.

**PODOBNE TEMATY**

Instrukcje if
Ustawianie zmiennych w AppleScript
Wartości stałe

**OTWÓRZ**

Przewodnik języka AppleScript - repeat

---



## Wartości stałe

Stała to zarezerwowane słowo w AppleScript, posiadające z góry określoną wartość. AppleScript zawiera różne stałe, których można użyć do uzyskania informacji o obiektach oraz dokonywania porównań lub obliczeń. Programy i inne obiekty mogą definiować własne stałe. Te stałe są zdefiniowane przez klasę definicji wartości.

Aby zobaczyć wartość stałej, wyklej ten przykład do Edytora AppleAcript, a następnie kliknij w Uruchom. Następnie kliknij

w Wynik, aby zobaczyć wartość stałej.

Wynikiem będzie również stała. Jest to stała skryptu, odnosząca się do wartości zwróconej przez polecenie.

```
get the current date
```

Oto kilka przykładów stałych:

- Stałe arytmetyczne: pi
- Stałe logiczne: true (prawda), false (fałsz)
- Stałe daty i czasu: minutes (minuty), hours (godziny), days (dni), weeks (tygodnie)
- Stałe tekstowe (łańcuchy): return, space (spacja), tab (tabulator)
- Stałe skryptowe: anything (cokolwiek), current application (bieżący program), it (to), me (stała zwrotna), missing value (brakująca wartość), my (stała zwrotna), result (wynik)

Więcej informacji o stałych można znaleźć w Przewodniku języka AppleScript.

**PODOBNE TEMATY**

Ustawianie zmiennych w AppleScript
Definicje obiektów
Definicje klas
Definicje zdarzeń

**OTWÓRZ**

Przewodnik języka AppleScript - stałe

---

**Polecenia podstawowe**

## Ustawianie zmiennych w AppleScript

Zmienna to kontener utworzony przez skrypt, służący do przechowywania jakiejś wartości. Podczas tworzenia zmiennej należy przypisać jej nazwę (identyfikator). Po uruchomieniu skryptu można umieścić wartość w zmiennej, użyć tej wartości, a także ją zmienić. Zmienne utrzymują swe wartości tylko w czasie działania skryptu. Po zamknięciu skryptu wartość zmiennej zostaje utracona.

- Poniższe przykłady ilustrują, jak określić unikatowy identyfikator oraz przypisać mu określoną wartość.

```
set myVar to "153"

set x to "Raporty kwartalne"

set x to false
```

Aby zobaczyć jak wartości są przechowywane w zmiennej, wklej ten przykład do Edytora AppleAcript, a następnie kliknij w Uruchom. Kliknij w Wynik, aby zobaczyć wartość stałej.

Więcej informacji o zmiennych można znaleźć w Przewodniku języka AppleScript.

**PODOBNE TEMATY**

Wartości stałe
Instrukcje if
Instrukcje repeat

**OTWÓRZ**

Przewodnik języka AppleScript - zmienne

---

**Polecenia podstawowe**

## Wskazywanie położenia programu

Programy w OS X są zbudowane hierarchicznie w pakiety. Pakiet to specjalny katalog, który zawiera katalogi i pliki, które łącznie składają się na program. Z tego powodu polecenia path to i info for mogą zwracać różne wyniki w przypadku programów.

Jeśli używasz polecenia path to wobec programu, na końcu ścieżki będzie zwykle .app. Rozszerzenia nazw plików nie zawsze są widoczne w Finderze, ale można nadal ich używać do identyfikacji pliku w skrypcie.

Jeśli użyjesz polecenia info for do uzyskania informacji o programie, właściwość Folder wyniku to zwykle będzie true. Możesz uzyskać więcej informacji o pliku i jego nazwie za pomocą dodatkowych właściwości w wyniku, takich jak extension hidden (ukryte rozszerzenie), name extension (rozszerzenie nazwy) lub displayed name (wyświetlana nazwa).

**PODOBNE TEMATY**

Polecenie path to
Polecenie info for
Odwoływanie się do katalogów, plików i dysków
Polecenie choose application
Polecenia skryptowania w Finderze

---

**Polecenia skryptowania Findera**

## Odwoływanie się do katalogów, plików i dysków

AppleScript pozwala odnosić się do niektórych katalogów, plików i dysków przy użyciu standardowych nazw, bez konieczności podawania pełnej ścieżki dostępu. AppleScript potrafi zidentyfikować położenie takich rzeczy jak katalog Dokumenty, biurko, katalog domowy i dysk startowy.

- Poniższy przykład pokazuje, jak odwoływać się w skrypcie do katalogu specjalnego, używając jego nazwy.

```
set x to the path to the desktop folder
```

**PODOBNE TEMATY**

Odwoływanie się do Findera w skryptach
Polecenie path to
Polecenie info for
Polecenia skryptowania w Finderze

## Odwoływanie się do Findera w skryptach

Aby użyć poleceń Findera w AppleScript, skieruj polecenia do Findera za pomocą instrukcji tell, w taki sam sposób, jak do innych programów.

- Oto przykład użycia instrukcji tell w skrypcie. Polecenie tell kieruje wynik funkcji zwracającej liczbę okien do obiektu Finder.app.

```
tell application "Finder.app"
        return the number of windows
end tell
```

**PODOBNE TEMATY**

Klasa Finder
Sposób przekazywania poleceń obiektom
Odwoływanie się do katalogów, plików i dysków
Definicje obiektów
Polecenia skryptowania w Finderze

**OTWÓRZ**

Podręcznik języka AppleScript - Specyfikatory obiektów



## Polecenia skryptowania w Finderze

Oto wybrane polecenia, których można używać z Finderem:

Polecenie activate

Klasa alias file

Klasa alias list

Klasa application file

Polecenie clean up

Polecenie close

Klasa column

Klasa computer-object

Klasa container

Polecenie copy

Polecenie count

Polecenie data size

Polecenie delete

Klasa desktop window

Klasa desktop

Klasa disk

Klasa document file

Polecenie duplicate

Polecenie eject

Polecenie empty trash

Polecenie exists

Klasa file

Klasa file information

Polecenie find

Klasa Finder

Klasa Finder window

Klasa folder

Polecenie get

Klasa icon view options

Klasa information window

Klasa internet location file

Klasa item

Klasa list view options

Polecenie make

Polecenie move

Polecenie moving folder window for

Polecenie open

Klasa package

Polecenie path to

Klasa preferences (katalog preferencji użytkownika)

Klasa preferences window

Polecenie print

Polecenie quit

Odwoływanie się do katalogów, plików i dysków

Odwoływanie się do Findera w skryptach

Polecenie reveal

Polecenie select

Polecenie set

Polecenie sort

Klasa trash (katalog kosza),

Polecenie update

Klasa window

---

**Polecenia skryptowania Findera**

## Standardowe polecenia dodatków skryptowych

Dodatki skryptowe uzupełniają zestaw przydatnych poleceń, których można użyć w skryptach.

Polecenie adding folder items to

Polecenie beep

Polecenie choose application

Polecenie choose color

Polecenie choose file

Polecenie choose file name

Polecenie choose folder

Polecenie choose from list

Polecenie choose URL

Polecenie clipboard info

Polecenie close access

Polecenie closing folder window for

Polecenie current date

Polecenie delay

Klasa dialog reply

Klasa display

Polecenie display dialog

Polecenie do shell script

Polecenie download

Klasa ftp item

Polecenie get eof

Polecenie info for

Polecenie list disks

Polecenie list folder

Polecenie load script

Polecenie mount volume

Polecenie offset

Polecenie open for access

Polecenie open location

Polecenie opening folder

Polecenie path to

Klasa POSIX path

Polecenie random number

Polecenie read

Polecenie round

Polecenie run script

Polecenie say

Polecenie scripting components

Polecenie set eof

Polecenie set the clipboard to

Polecenie set volume

Polecenie store script

Polecenie summarize

Polecenie system attribute

Polecenie time to GMT

Polecenie upload

Klasa url

Polecenie write

---